**Developing a Predictive Machine Learning Model**

**Emander D. Dangla, Wonyoung Chung, Utku Emecan, Nilkanth R. Patel**

**School of Engineering Technology and Applied Science, Centennial College**

**COMP 247: Supervised Learning**

**Prof. Merlin J. R. Dennis**

**April 16, 2023**

Table of Contents

[Executive Summary 3](#_Toc132553824)

[Overview of Solution 4](#_Toc132553825)

[Data Exploration and Findings 6](#_Toc132553826)

[Data Exploration 6](#_Toc132553827)

[Findings 13](#_Toc132553828)

[Tools Used 13](#_Toc132553829)

[Feature Selection 14](#_Toc132553830)

[Findings 15](#_Toc132553831)

[Data Modeling 16](#_Toc132553832)

[Model Building 18](#_Toc132553833)

# Executive Summary

In this project, our objective was to predict the likelihood of fatal cases in traffic collisions in Toronto using the KSI dataset, which includes information about all traffic collision events where a person was either Killed or Seriously Injured from 2006 to 2021. To achieve this, our team has developed and compared four machine learning models - Random Forest, Histogram-based Gradient Boosting, Logistic Regression, and Support Vector Machines (SVM) with Bagging - to predict fatal traffic collision cases in Toronto. Our DataLoader class loads the dataset and performs data cleaning, transformation, standardization, SMOTE sampling, feature selection, and encoding.

We evaluated the performance of the models using accuracy, precision, recall, and F1-score metrics. The Histogram-based Gradient Boosting model demonstrated the best performance, with an accuracy of 96.46%, a precision of 96.42%, a recall of 96.46%, and an F1-score of 96.31%.

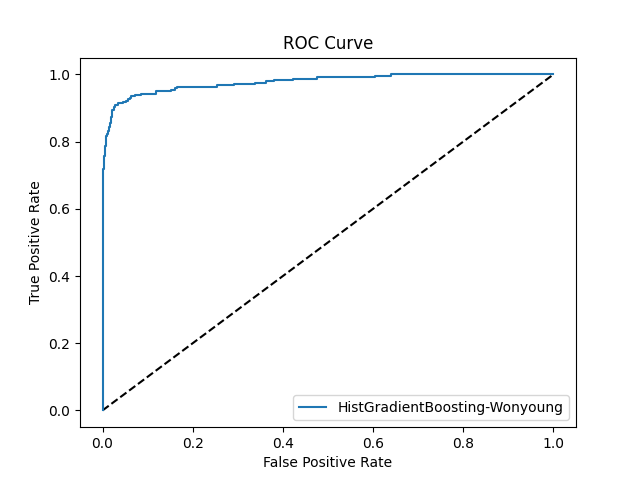
# Overview of Solution

Our solution involved the creation of a DataLoader class to efficiently manage the KSI dataset, preprocess the data, and prepare it for modelling. This class was responsible for loading the dataset, cleaning the data, transforming features, selecting relevant features, applying SMOTE sampling, standardizing the data, and encoding categorical variables.

We then built four machine learning models - Random Forest, Histogram-based Gradient Boosting, Logistic Regression, and SVM with Bagging - to predict whether a traffic collision would result in a fatality. We trained each model on the preprocessed data and evaluated it using accuracy, precision, recall, and F1-score metrics. The performance of each model was as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1 Score |
| Random Forest | 92.51% | 92.07% | 92.51% | 91.96% |
| Histogram-based Gradient Boosting | 96.46% | 96.42% | 96.46% | 96.31% |
| Logistic Regression | 61.52% | 80.75% | 61.52% | 67.41% |
| SVM With Bagging | 61.92% | 80.52% | 61.92% | 67.73% |

[Chart, line chart

Description automatically generated](https://centennialcollegeedu-my.sharepoint.com/:w:/g/personal/wchung34_my_centennialcollege_ca/ESCMvl3WDspAhL3NUwFQ1UABPBsm-BM9HRJBW5h9NlJMUQ)

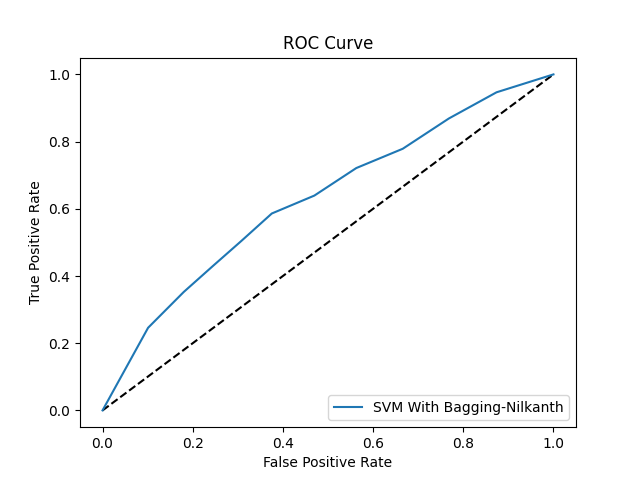
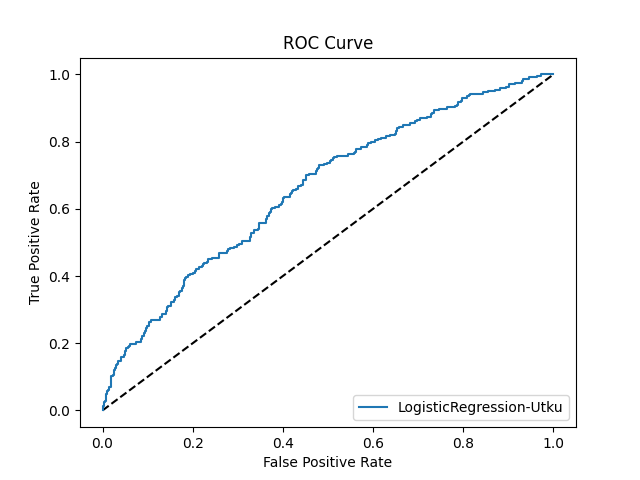


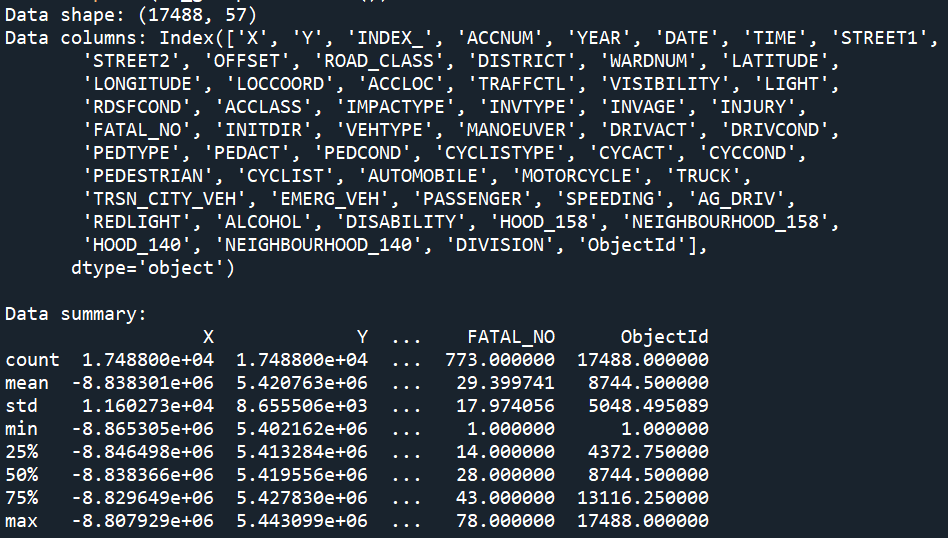
Table 1 Confusion Matrices

|  |  |  |  |
| --- | --- | --- | --- |
| Random Forest | Hist Gradient Boosting | Logistic Regression | SVM With Bagging |
| [[1475 30]  [ 101 143]] | [[1495 10]  [ 52 192]] | [[929 576]  [ 97 147]] | [[940 565]  [101 143]] |

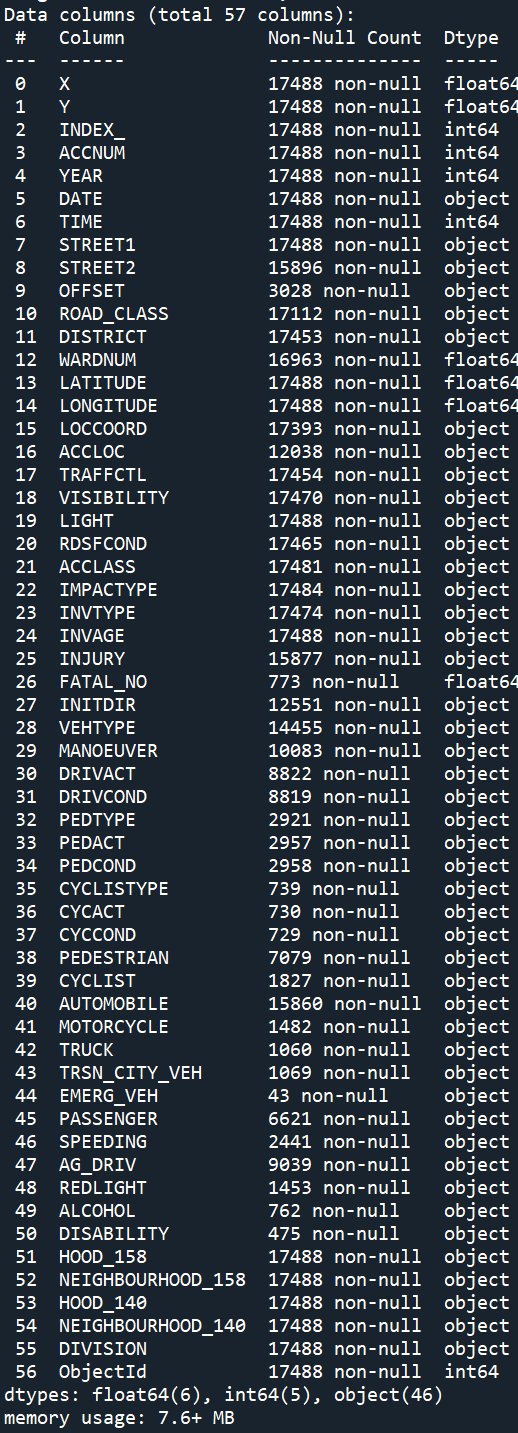
Based on these results, the Histogram-based Gradient Boosting model outperformed the other models, making it the most suitable choice for predicting fatal traffic collisions in Toronto using the KSI dataset.

# Data Exploration and Findings

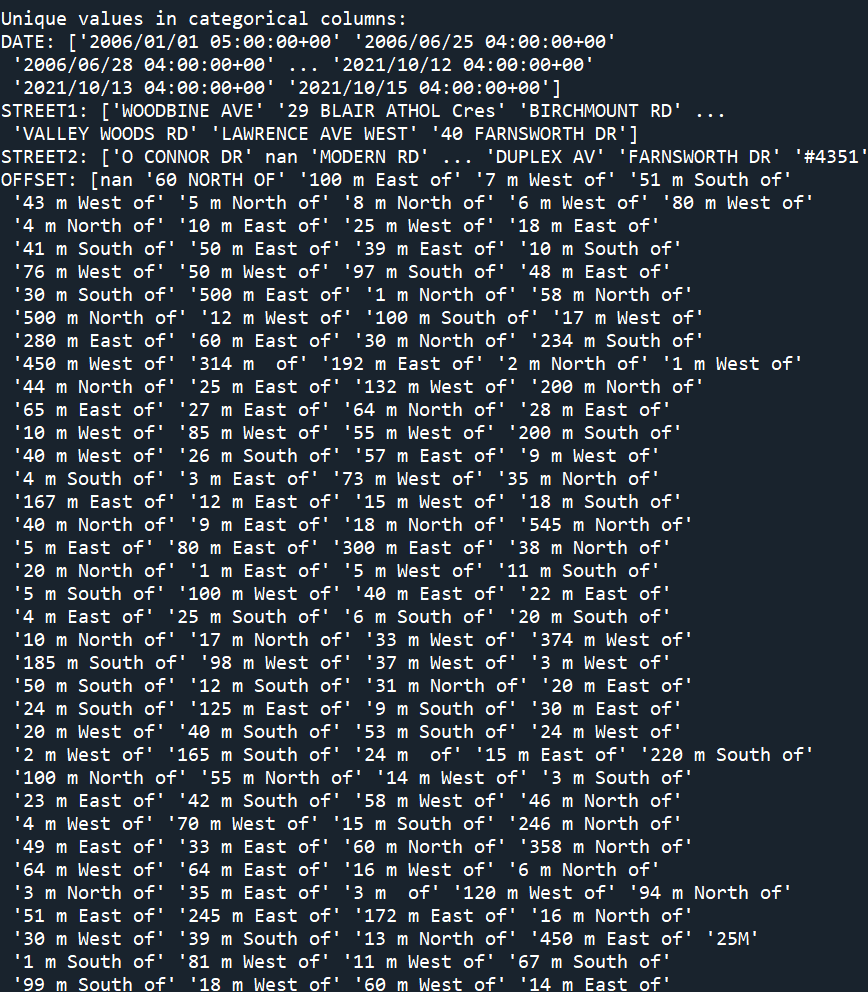
## Data Exploration



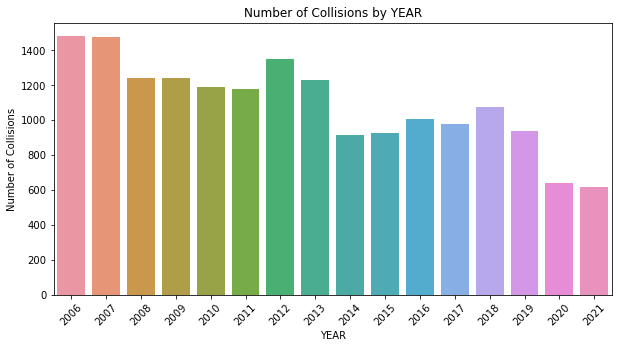
57 columns exist with 17488 rows and not much to find on this part of exploration.

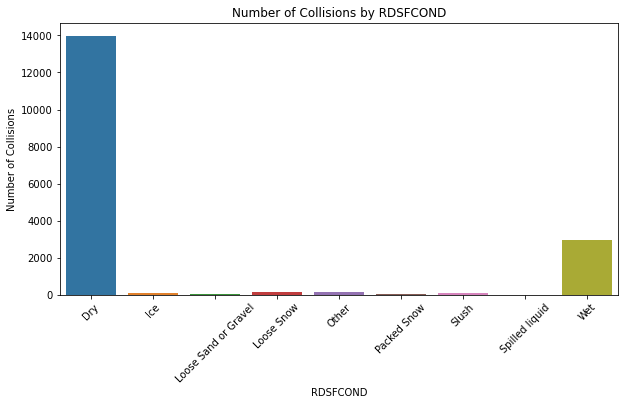


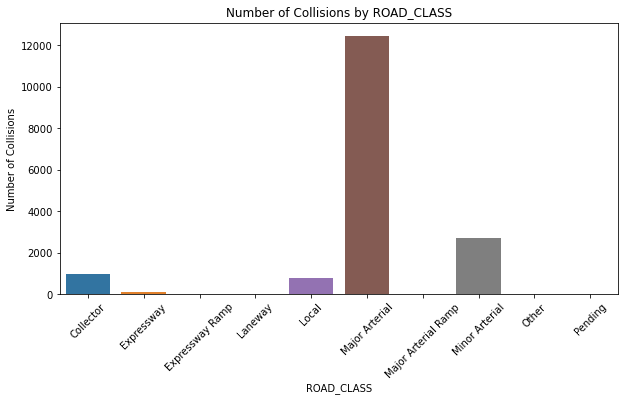
The Dataset is a mixture of numerical and categorical fields, and we can see the rows that have a lot of null values.

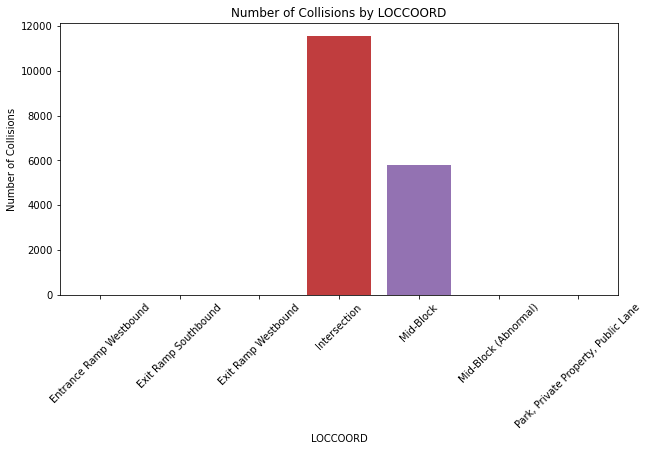


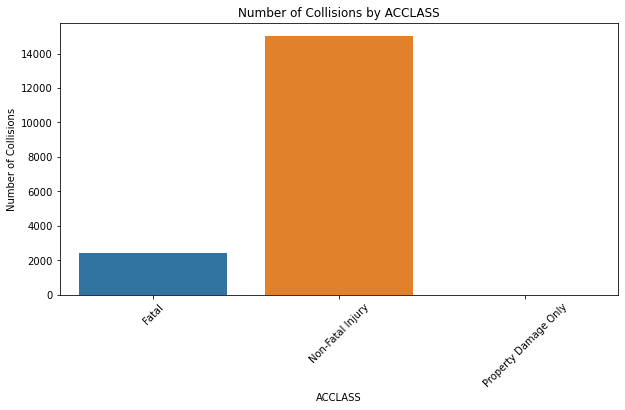
Here we can see which fields have a lot of unique values so we can remove them from the feature selection.

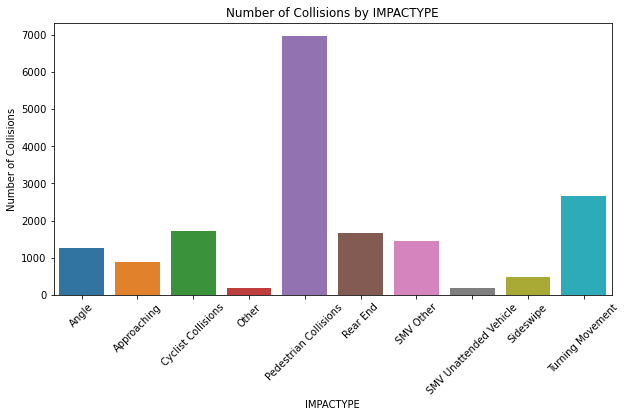


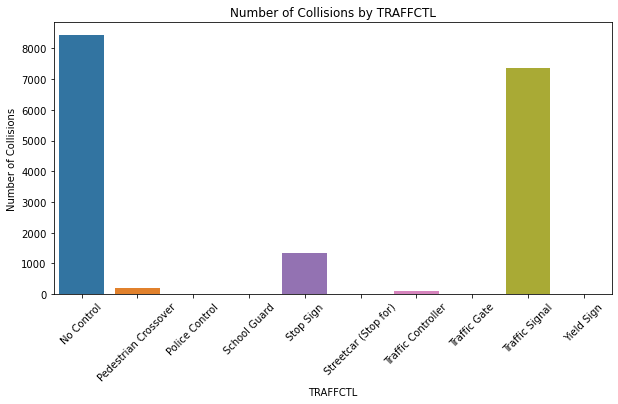


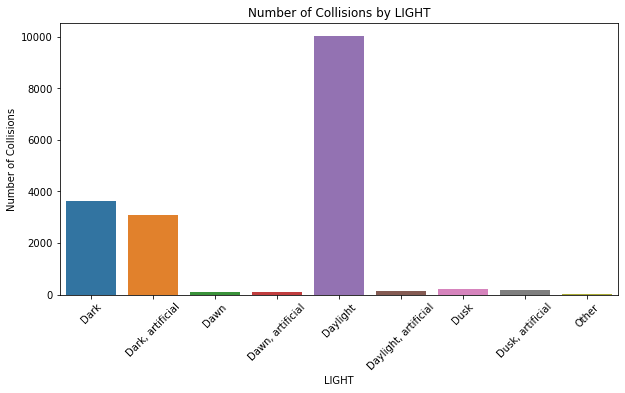












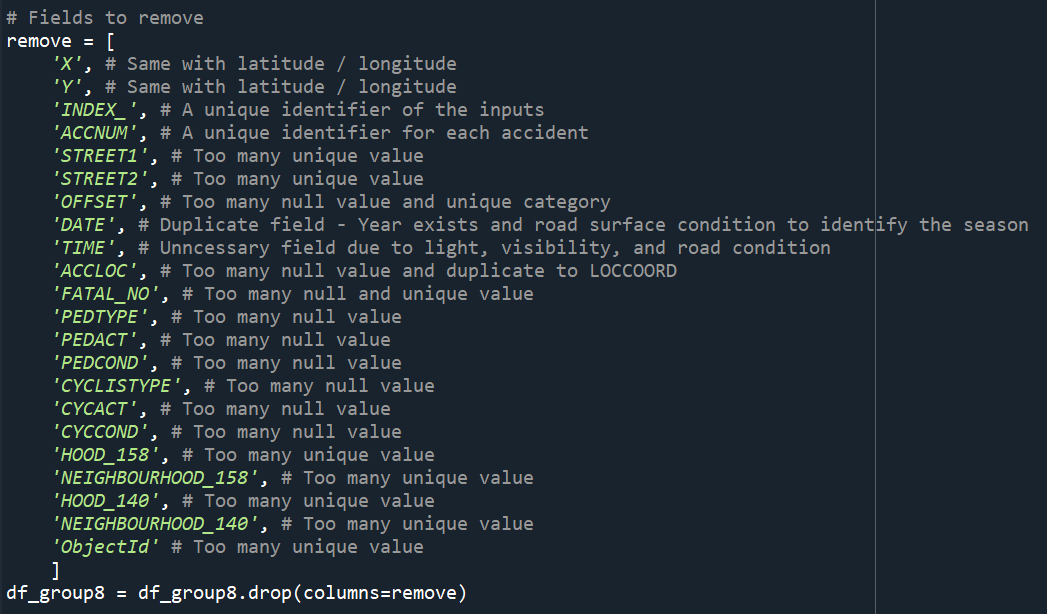
## Findings

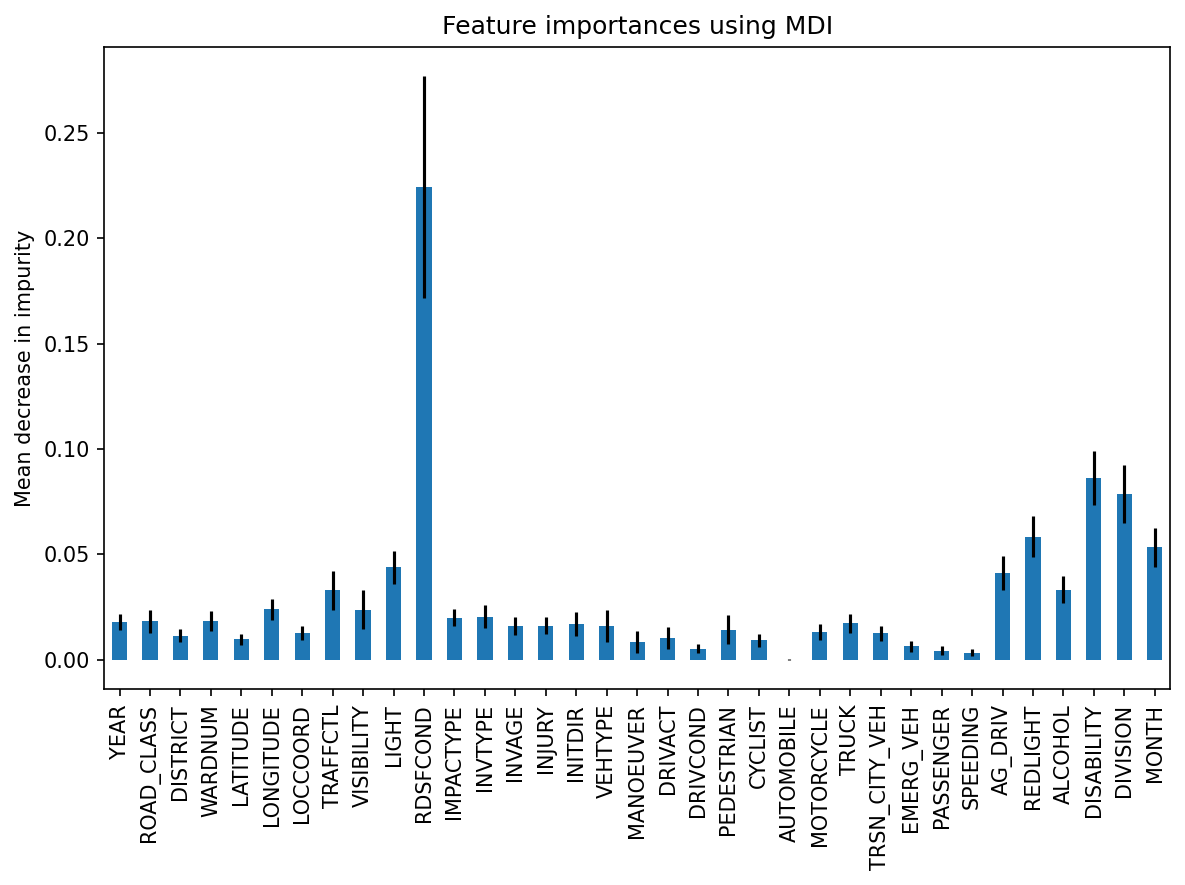
We found out that 90% of the collision is non-fatal and around 10% is fatal. We also discovered that there are some fields that can be a red herring for the model prediction such as Injury fields which contains major and minor classes. There are also a lot of yes or no features that can be fixed by populating the null value with no. In addition, there is another category which is property damage that can be non-fatal.

## Tools Used

We used matplotlib and histogram.

# Feature Selection

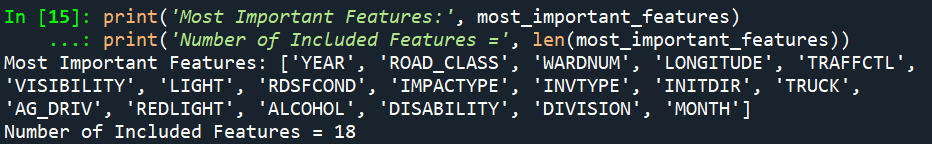




We have used Random Forest’s feature importance function to identify the important features amongst the categorical and numerical features.

## Findings

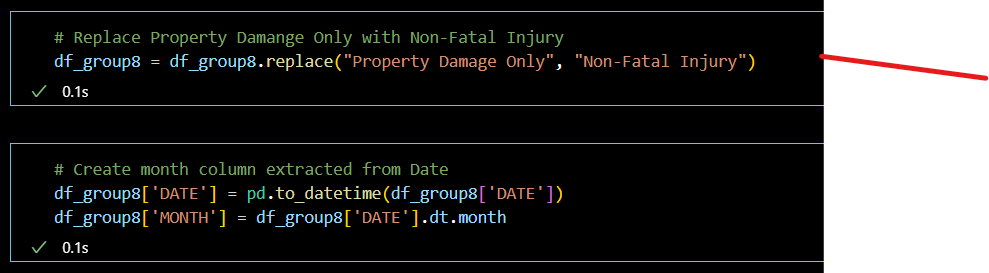
It shows that the road condition is the most important feature, followed by disability, division, month, red light, aggressive driver, and light.



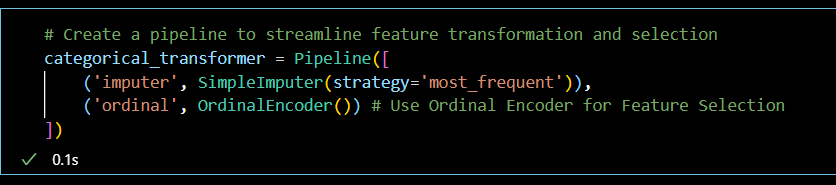
We have filtered this by selecting only the features with importance greater than 0.017. As you can see in the screenshot above. Only 18 features remain.

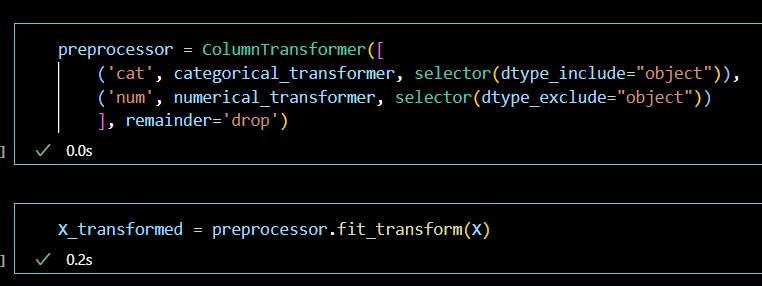
# Data Modeling

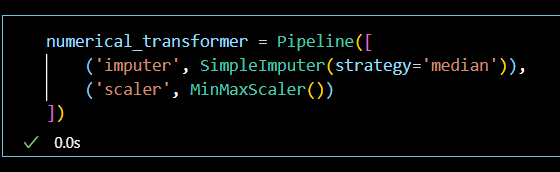
In this project, we utilized the KSI dataset containing traffic collision events that resulted in either fatalities or severe injuries in Toronto from 2006 to 2021. The DataLoader class was responsible for loading the dataset, cleaning it, transforming it, selecting essential features, and preparing the data for modelling. The data cleaning involved handling missing values, extracting new features, and dropping unnecessary columns. The new features, such as "MONTH" and "DAY\_WEEK," were derived from the "DATE" column.



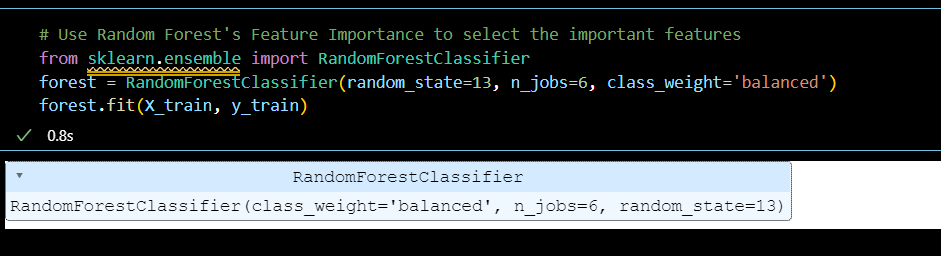
We created a column transformation pipeline to streamline feature preprocessing, which included imputing missing values, scaling numerical features, and encoding categorical features.

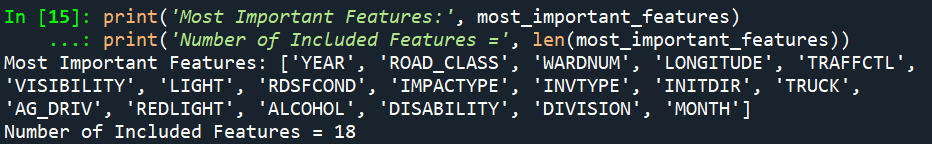






After preprocessing the dataset, we used a Random Forest classifier to determine the most important features, which were then used for further analysis.





# Model Building

The Model Building phase involved training and evaluating various classifiers on the preprocessed and feature-selected dataset. The DataLoader class was responsible for splitting the dataset into training and testing sets, ensuring a balanced class distribution by applying the Synthetic Minority Over-sampling Technique (SMOTE) to the training set, and saving the data for subsequent model training. We built four models to predict fatal cases: Random Forest, Histogram-based Gradient Boosting, Logistic Regression, and Support Vector Classifier (SVC) with bagging. We evaluated these models based on their performance metrics, such as the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). By comparing the performance of the different models, we aimed to identify the most effective model for predicting traffic collision outcomes in the city of Toronto.

## Data Sources

* **Train Data:** We used a dataset containing historical customer information from the telecom company, including demographic, billing, and usage data. The dataset consists of 15,000 records with a churn rate of 20%.
* **Test Data:** A separate dataset of 5,000 records was used for evaluating model performance. This dataset exhibits a similar churn rate as the training data.

## Model Building Assumptions

* The historical data is representative of future customer behavior.
* The train and test datasets are assumed to come from the same population.
* The features included in the dataset have a direct or indirect impact on customer churn.
* The churn rate in the historical data is representative of the true churn rate.

## Data Preprocessing

* Missing values were imputed with median values for continuous variables and mode values for categorical variables.
* Categorical variables were one-hot encoded to create dummy variables.
* Feature scaling was applied to normalize continuous variables.
* The train dataset was split into a training set (80%) and a validation set (20%) using stratified sampling to maintain the same churn rate across all subsets.

## Algorithms Tested

* Logistic Regression
* Decision Trees
* Random Forest
* Support Vector Machines
* Neural Networks

## Model Evaluation Metrics

* Accuracy
* Precision
* Recall
* F1-score
* Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC)